Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Journal of Biomedical Informatics 123 (2021) 103918

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

Original Research

Query bot for retrieving patients' clinical history: A COVID-19 use-case

Yibo Wang [a,*], Amara Tariq [d], Fiza Khan [c], Judy Wawira Gichoya [c,b], Hari Trivedi [c,b], Imon Banerjee [d,e]

[a] Department of Computer Science, Emory University, Atlanta, GA 30322, USA
[b] Department of Biomedical Informatics, Emory School of Medicine, Atlanta, GA 30322, USA
[c] Department of Radiology, Emory School of Medicine, Atlanta, GA 303224, USA
[d] Department of Radiology, Mayo Clinic, Arizona, AZ 85054, USA
[e] School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, AZ 85281, USA

ARTICLE INFO

Keywords:
Information retrieval
Clinical notes
BERT
k-means
Relevance feedback

ABSTRACT

Objective: With increasing patient complexity whose data are stored in fragmented health information systems, automated and time-efficient ways of gathering important information from the patients' medical history are needed for effective clinical decision making. Using COVID-19 as a case study, we developed a query-bot information retrieval system with user-feedback to allow clinicians to ask natural questions to retrieve data from patient notes.

Materials and methods: We applied clinicalBERT, a pre-trained contextual language model, to our dataset of patient notes to obtain sentence embeddings, using K-Means to reduce computation time for real-time interaction. Rocchio algorithm was then employed to incorporate user-feedback and improve retrieval performance.

Results: In an iterative feedback loop experiment, MAP for final iteration was 0.93/0.94 as compared to initial MAP of 0.66/0.52 for generic and 1./1. compared to 0.79/0.83 for COVID-19 specific queries confirming that contextual model handles the ambiguity in natural language queries and feedback helps to improve retrieval performance. User-in-loop experiment also outperformed the automated pseudo relevance feedback method. Moreover, the null hypothesis which assumes identical precision between initial retrieval and relevance feedback was rejected with high statistical significance ($p \ll 0.05$). Compared to Word2Vec, TF-IDF and bioBERT models, clinicalBERT works optimally considering the balance between response precision and user-feedback.

Discussion: Our model works well for generic as well as COVID-19 specific queries. However, some generic queries are not answered as well as others because clustering reduces query performance and vague relations between queries and sentences are considered non-relevant. We also tested our model for queries with the same meaning but different expressions and demonstrated that these query variations yielded similar performance after incorporation of user-feedback.

Conclusion: In conclusion, we develop an NLP-based query-bot that handles synonyms and natural language ambiguity in order to retrieve relevant information from the patient chart. User-feedback is critical to improve model performance.

1. Introduction

With the recent COVID-19 pandemic [1], there has been a surge in patient volumes at many hospitals and emergency departments, requiring quick and accurate information retrieval for these patients to maximize quality and efficiency of care. This information cannot always be obtained directly from the patient or caregiver, but rather must be obtained from clinical records, which are frequently disjointed [2]. Moreover, clinical records are unstructured and the recording style differs for different types of clinical notes. Many terminologies in clinical notes are synonymous, such as 'high blood pressure' and 'hypertension'. Moreover, some symptoms and therapies may directly or indirectly represent certain diseases [3,4]. For example, the presence of 'insulin' usually indicates that the patient has diabetes. These complexities necessitate a natural language processing (NLP) based method of information retrieval and contextualization.

* Corresponding author.
E-mail address: imon.banerjee@asu.edu (Y. Wang).

https://doi.org/10.1016/j.jbi.2021.103918
Received 21 December 2020; Received in revised form 17 September 2021; Accepted 19 September 2021
Available online 21 September 2021

The general information retrieval process is conceptualized as: 1) the user has an information need and forms a query; 2) user sends the query to information retrieval system; 3) information retrieval system returns top-ranked results; 4) user evaluates the results and decides whether to continue or stop. Traditional information retrieval methods include probabilistic models like the binary independence model which assumes that sentences are binary vectors, and BM25 model [5] which is a bag-of-words [6] retrieval function based on term frequency – inverse document frequency (TF-IDF). These methods are based on query terms appearing in sentences, regardless of their synonyms and proximity within sentences. Vector space models [7] form another set of information retrieval models that consider sentences and queries as vectors and use similarity between the query and sentence as a scale value. Other information retrieval methods like learning-to-rank [8] apply machine learning techniques to information retrieval systems.

With the widespread use of electronic health records (EHRs), there have been many works on information retrieval from clinical reports. StarTracker [9], which is a web-based Boolean retrieval search engine, allows users to search an existing EHR system for panels of patients based on specific diagnostic, demographic and clinical criteria. Electronic Medical Record Search Engine (EMERSE) [10] is another web-based Boolean medical search engine designed that also handles problems like spelling errors and query recommendations. Besides traditional approaches, a convolutional neural network is trained to predict patients' diagnostic ICD codes from MIMIC-III database in [11], and then activations from fully connected hidden layer are used as dense representation of the text to improve accuracy. Existing clinical search engines focus primarily on the single arm of the information retrieval challenge – single shot precision, however computational efficiency and importance of user feedback has not been considered in the previous studies.

To improve information retrieval models over time, user feedback must be gathered on the relevance of returned results [12]. For example, Google SearchWiKi allows users to annotate and reorder search results, which are then incorporated into future results returned by the engine. Pseudo relevance feedback [13] is one of the most popular relevance feedback models. It assumes that the top k results of the initial retrieval are relevant, and adjusts query depends on this assumption. Pseudo relevance feedback has been proven to be effective in many tasks. However, since the top k results are not necessarily relevant, pseudo relevance feedback sometimes leads the query towards a wrong direction. Instead of pseudo relevance feedback, we use real user feedback to improve results that are much more reliable.

In this paper, we built an interactive retrieval platform for retrieving clinical history of COVID-19 patients using *natural language query* and developed a graphical user interface for convenience. Contrast with the existing clinical query systems, we also use relevance feedback to improve performance, raising precision by 27%/42%, and use K-Means to shorten computation time, making our model feasible for practical real-time applications in a standard machine available for clinical usage.

2. Method

Fig. 1 shows the flow of our model. We fine-tune clinicalBERT to obtain a language model for clinical text space that is later used to generate sentence embeddings. Then we use K-Means to reduce computation time through cosine-similarity based clustering for real-time feedback implementation, and Rocchio algorithm [14] to improve retrieval performance. The core processing blocks are described in the following section.

2.1. Cohort building and data preprocessing

With the approval of Emory University Institutional Review Board, we retrieved all the clinical notes of patients who were COVID-19 positive across 12 centers of Emory Healthcare. A total of 1,688 patients confirmed to be COVID-19 positive between 01/01/2020 to 06/29/2020 were retrieved (55.1% female, 19.4% White, 48.6% African American). We collected current and historical encounter data for these patients from 01/02/2018 to 06/29/2020. A confirmed COVID-19 diagnosis was defined as either a positive SARS-CoV-2 RNA detection test [15] or a diagnosis code for COVID-19 (ICD-10 U07.1). Among them 666 patients (39.4%) were hospitalized and 392 patients (23.2%) were transferred to ICU. We retrieved all free-text clinical notes for each patient for one year preceding their COVID-19 diagnosis, with a mean of 1,576 notes per patient, 22 sentences per note and 42 words per sentence. The distribution of the types of clinic notes is shown in Fig. 2.

As the first pre-processing step, we split clinical notes into sentences, and discarded sentences with fewer than 4 terms or>200 terms. We observe that very long sentences are usually tabular data from the EHR template and do not have semantically meaningful information when concatenated as strings. Thereafter, we used standard NLP processes to clean the sentences, including lowercasing, deleting punctuations and removing stopwords including terms like 'patient', 'have', 'is' and others. After the cleaning, our cohort contained a total of 658,327 sentences with a mean of 21 words.

2.2. Bidirectional encoder representations from transformers (BERT) vectorization

BERT [16] architecture uses bi-directional transformers and generates contextualized word representations by training a masked language model. It has been proven to be one of the most powerful natural language processing models to date, and had improved the state-of-art

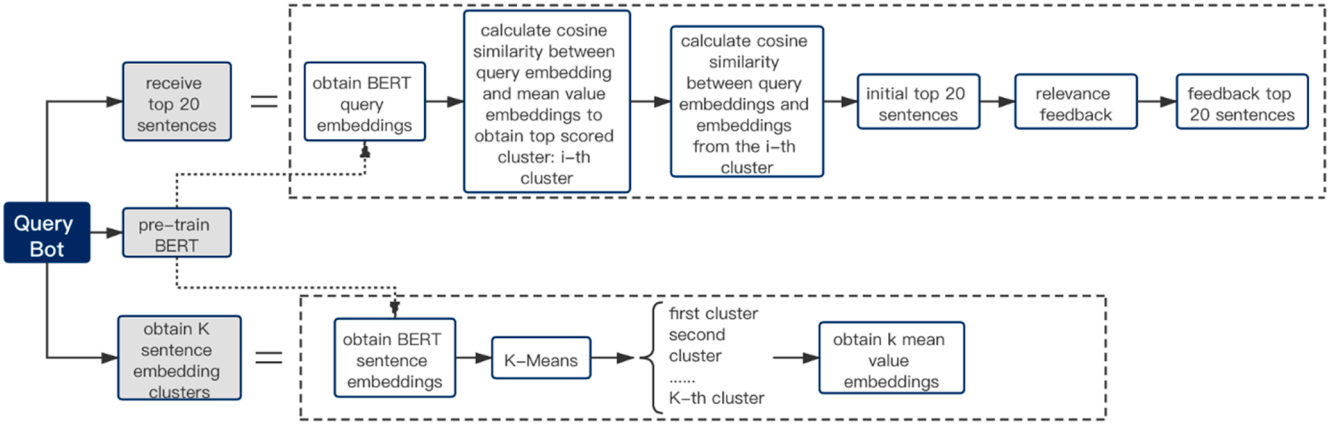

Fig. 1. Flowchart of our QueryBot model showing each representative module and output. Arrows highlights information exchange between modules. Gray boxes are summaries of each core processing module - (1) language space fine-tuning; (2) clustering for real-time processing of queries; (3) ranked retrieval.

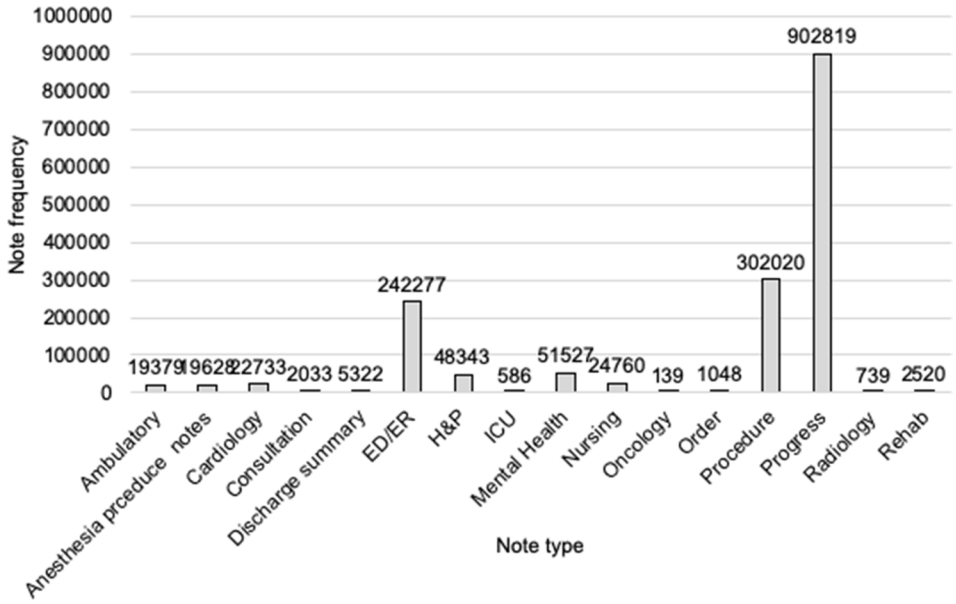

Fig. 2. The distribution of the types of clinical notes extracted for one year preceding COVID-19 diagnosis for 1688 patients who were COVID-19 positive. There are 179 types of notes for all patients.

performance on at least 11 tasks when it was published. BERT is pre-trained on two unsupervised learning tasks, masked language model and next sentence prediction. A BERT model called clinicalBERT [17], was trained on PubMed abstracts (PubMed), [18] Central full-text articles (PMC) and all MIMIC notes (implementation available on https://huggingface.co/emilyalsentzer/Bio_ClinicalBERT). We use masked language modeling tasks to fine-tune pretrained on COVID-19 clinical note data, so that the model can learn specific domain knowledge regarding COVID-19. Multiple application-dependent strategies can be applied to generate BERT sentence embeddings. For example, [CLS] embedding is used for next sentence prediction tasks. According to [19], average BERT embedding, which is the average token embeddings of the last hidden layer, has better performance than [CLS] embedding in textual similarity tasks. Therefore, we average the last hidden layer token representation of BERT as sentence embedding, which is ultimately a 768-dimension vector.

For comparison, we also included the Word2Vec model in our experiments. Our Word2Vec model is finetuned using our patient notes dataset based on https://bio.nlplab.org, which are induced from PubMed and PMC texts and their combinations.

2.3. Query retrieval

Cosine similarity between the BERT sentence embedding is defined as:

$$cosine\ similarity\ = (A \cdot B)\ /(||A||\cdot||B||)$$

where *cosinesimilarity* a measure of similarity between two sentence embeddings (A and B) based on the cosine of angle generated by the embeddings in the vector spaces, and therefore represents orientation of embeddings rather than magnitude. Since we are using vectors to represent sentences, we use cosine similarity to measure the relationship between queries and sentences. Higher cosine similarity score means the query-sentence pair is more relevant.

However, since our dataset contains approximately 700,000 sentences with 768-dimension embedding, it takes over 20 min to calculate one query using a standard machine with 32 CPU cores (2 AMD Opteron 6376 processors) and 128 GB RAM. This magnitude of delay is of no value in practical applications which need real-time information retrieval, especially when interaction with the emergency care-team is involved. To shorten the computation time and improve the clinical adoption, we use K-Means to cluster all the sentences into K clusters first, where K is selected by gap statistics (optimal $k = 99$). K-Means works based on the assumption that sentences that describe similar semantic information should have close numeric embeddings and thus will be grouped into the same cluster. Fig. 3 visualizes 5 randomly selected clusters and reveals the reliability of clustering performance. After obtaining K clusters, we calculate cosine similarity between the query and the mean embedding of each cluster, and select the highest-scoring cluster as candidate sentences set. Then we calculate cosine similarity score between query and candidate sentences to retrieve the top 20 sentences as initial retrieval. While it requires 20 min to calculate 1 query, K-Means helps to shorten the computation time to less than 30 s.

2.4. User feedback

Even though contextual embeddings should capture the semantics of the query and retrieved sentences, the final similarity evaluation depends heavily upon the perception of the end-users [21]. Relevance

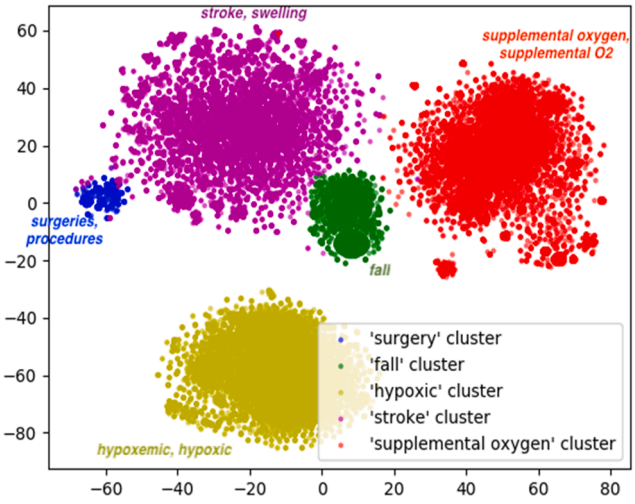

Fig. 3. Visualization of five randomly selected clusters projected in 2D using t-SNE [20].

feedback depends on the assumption that keeping users in loop to capture the perception will help gather user feedback and revise the language space specifically for individual queries. For example, the initially retrieved results of '*What is the surgical history of the patient?*' include both 'medical history' and 'surgical history' since wordings are similar. If users select relevant sentences and deselect irrelevant sentences, then the language space will be adjusted based on the user-feedback information and more relevant sentences will be retrieved. As shown in Fig. 4, after relevance feedback, the query approaches relevant sentences and departs from irrelevant sentences in the language space. There are many relevance feedback models, like pseudo relevance feedback and probabilistic relevance feedback. Here we use Rocchio algorithm [14] which updates the query based on feedback as follows:

$$q_{opt} = \alpha \cdot q_0 + \beta \cdot \frac{1}{C_R} \cdot \sum_{d_i \in C_R} C_R - \gamma \cdot \frac{1}{C_{IR}} \cdot \sum_{d_j \in C_{IR}} C_{IR}$$

where q_0 is initial query, q_{opt} is feedback query, C_R is the relevant sentence set selected by users and C_{IR} is the irrelevant sentence set. We set $\alpha = 1, \beta = 0.75, \gamma = 0.25$ based on empirical testing. However, such feedback needs strong human involvement for every retrieval. Thus, we compare Rocchio algorithm with automated pseudo relevance feedback which directly assumes top 10 sentences are relevant and others are irrelevant and continue the iteration to update the query space.

2.5. Interface design

We use the *tkinter* library [22] of Python 3.6 to build a working prototype of the interface which allows the interaction with the BERT language model via query formulated in natural language and collects user feedback for each query, as shown in Fig. 5. After choosing relevant sentences and clicking on the 'Complete' button, the feedback information will be collected and used to update the retrieved results in the next round. The interface allows exploration of both population-level data retrieval (study trends for all patients) and queries for individual patients. Users can use both query and patient ID to filter required information. Demo video of the system can be found here – https://youtu.be/GYpMBGHy080.

3. Results

We validate the proposed query platform by collecting 17 general natural language queries and 9 COVID-19 specific queries, relevant for retrieval of patients' clinical history, from two emergency medicine physicians. For the testing with Rocchio algorithm, we evaluate our system independently for each reviewer as the algorithm is sensitive to user feedback. We recruited two radiologists trained in Emergency Medicine and with at least 4 years of ED experience, as users of our system. Model performance was evaluated for top-20 retrieved sentences for three rounds where in round 1, there was no feedback was incorporated and starting from round 2 the retrieval space was modified with user feedback. (see Table 1 for general queries and Table 2 for COVID-19 related queries).

We compared the Rocchio algorithm with pseudo relevance feedback (see Table 3 for general queries and Table 4 for COVID-19 specific queries). We also calculated *Mean Average Precision* (MAP) [23], which is defined as:

$$MAP = \frac{\sum_{q=1}^{Q} AveP(q)}{Q}$$

where Q is the number of queries and *AveP* is average precision (i.e. the mean of the precision scores in the ranked retrieval). MAP results are visualized in the bar chart in Fig. 6.

To evaluate the significance of feedback on model performance, we perform a two-sided T-test comparing performance in round 1 to performances in rounds 2 and 3. We consider the null hypothesis (H_0) that baseline (round 1) has an identical expected precision value as rounds 2 and 3 following feedback and report the p-values in Table 4. As seen from Table 4, the null hypothesis has always been rejected with significant confidence when baseline is compared with the subsequent round of feedback which shows that user feedback impose statistically significant performance improvement.

Besides, we compared clinicalBERT with other popular word embedding methods, like Word2Vec, TF-IDF and bioBERT. MAP of all of these embeddings are shown in Tables 6 and 7 for general and COVID-19 specific queries respectively, with feedback of user 1 incorporated through the Rocchio algorithm.

For deep investigation of our model's performance, we deliberately select queries with the same meaning but different expressions, and test whether our model can handle the ambiguity in natural language queries. Selected queries include pairs like 'Does the patient have any surgeries?' and 'What is surgical history of the patient?', 'Has patient had a stroke?' and 'Does patient history of stroke?' and so on. For most of the similar query pairs, although the retrieved results may be different in the first round (see Supplementary Table 1), the results are very similar in the final round after relevance feedback. Our query set was also purposefully selected to include some synonyms examples. For example, 'hypertension' and 'high blood pressure', 'hypoxic' and 'requiring supplemental oxygen', 'shortness of breath' and 'having difficulty breathing', and so on. Our results show that clinicalBERT can handle issues of clinical synonyms because sentences like 'Severe

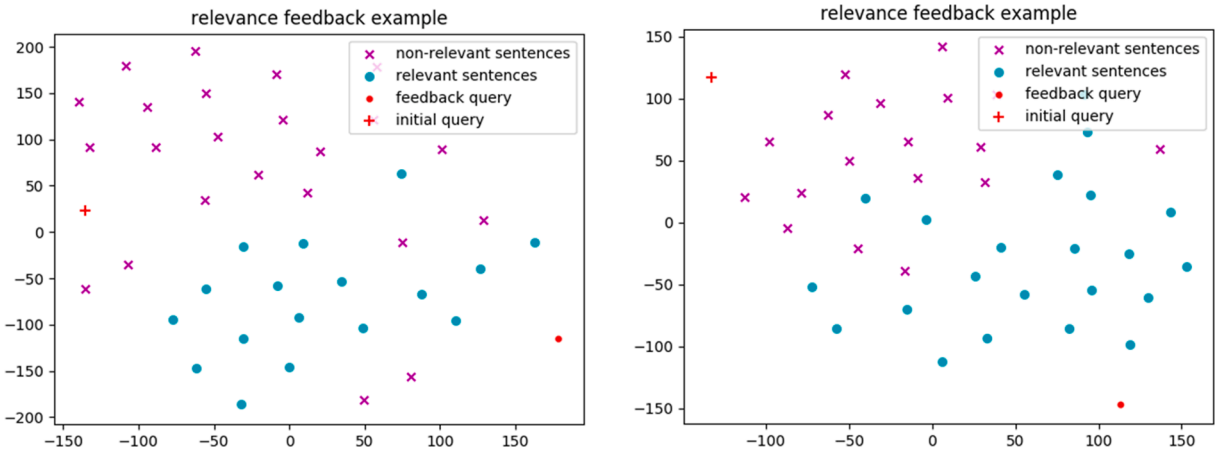

Fig. 4. t-SNE projection of the language space with relevant and non-relevant sentences along with query vectors before and after user feedback. Queries: (left) '*What is surgical history of the patient?*', (right) '*Has patient had a stroke?*'

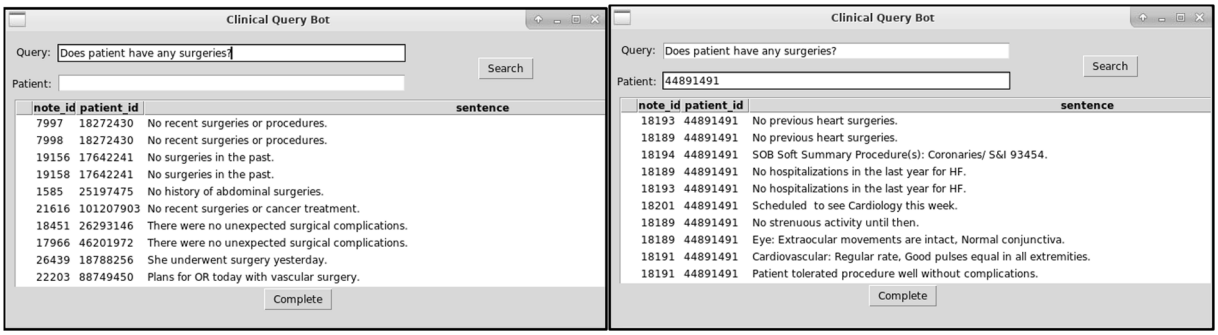

Fig. 5. Sample results of the user interface using the query 'Does patient have any surgeries?' Retrieved results are shown at the population-level (left) and for an individual patient (right).

Table 1
Precision for general queries across rounds 1, 2, and 3 for both reviewers of Rocchio algorithm. The overall MAP is listed at the end, demonstrating a clear increase in MAP by the third round, showing the model is improving by incorporating user feedback.

Rocchio Algorithm (General Queries)	KMeans (k = 99), reviewer 1				KMeans (k = 99), reviewer 2			
	Round1	Round2	Round3	Mean	Round1	Round2	Round3	Mean
Does patient have any surgeries?	0.75	0.79 (15/19)	0.79 (15/19)	0.78	0.80	1.00 (16/16)	1.00 (16/16)	0.93
What is surgical history of the patient?	0.10	0.82 (14/17)	0.83 (15/18)	0.58	0.10	0.72 (13/18)	0.83 (15/18)	0.55
Does patient have diabetes?	0.95	1.00 (20/20)	1.00 (20/20)	0.98	0.25	0.37 (7/19)	1.00 (7/7)	0.54
Is patient diabetic?	0.40	0.80 (8/10)	0.89 (8/9)	0.70	0.55	1.00 (15/15)	1.00 (15/15)	0.85
Does patient have hypertension?	1.00	0.95 (19/20)	1.00 (20/20)	0.98	0.35	1.00 (11/11)	1.00 (11/11)	0.78
Does patient have high blood pressure?	0.20	0.60 (6/10)	0.88 (7/8)	0.23	0.05	0.31 (5/16)	0.42 (5/12)	0.26
Does patient have diabetes and hypertension?	1.00	1.00 (20/20)	1.00 (20/20)	1.00	0.55	1.00 (19/19)	1.00 (14/14)	0.85
Has patient had a stroke?	0.25	0.85 (17/20)	0.85 (17/20)	0.65	0.15	0.85 (17/20)	0.9 (18/20)	0.63
Does patient history of stroke?	0.10	0.74 (14/19)	0.74 (14/19)	0.53	0.05	0.17 (2/12)	1.00 (12/12)	0.41
Did patient have a fall?	0.85	1.00 (18/18)	1.00 (18/18)	0.95	0.85	1.00 (18/18)	1.00 (18/18)	0.95
Is patient hypoxic?	1.00	1.00 (20/20)	1.00 (20/20)	1.00	1.00	1.00 (20/20)	1.00 (20/20)	1.00
Is the patient's SpO2 ≤ 94% on room air?	0.95	1.00 (20/20)	1.00 (20/20)	0.98	1.00	1.00 (20/20)	1.00 (20/20)	1.00
Is the patient requiring supplemental oxygen?	0.55	0.80 (12/15)	1.00 (12/12)	0.78	0.55	0.63 (12/19)	1.00 (12/12)	0.73
Does the patient have an abnormal chest X-ray?	0.95	1.00 (20/20)	1.00 (20/20)	0.98	0.90	1.00 (20/20)	1.00 (20/20)	0.97
Is the patient requiring mechanical ventilation?	0.80	1.00 (20/20)	1.00 (20/20)	0.93	0.60	1.00 (18/18)	1.00 (18/18)	0.87
When did the patient's symptoms start?	0.90	0.90 (18/20)	1.00 (20/20)	0.58	0.60	0.90 (18/20)	1.00 (20/20)	0.83
When was the patient's last hospitalization?	0.50	1.00 (13/13)	1.00 (13/13)	0.83	0.50	1.00 (13/13)	1.00 (13/13)	0.83
Overall MAP	0.66	0.90	0.94	0.83	0.52	0.82	0.95	0.76

Table 2
Precision for COVID-19 specific queries across rounds 1, 2, and 3 for both reviewers of Rocchio algorithm. The overall MAP is listed at the end, demonstrating a clear increase in MAP by the third round, showing the model is improving by incorporating user feedback.

Rocchio Algorithm (COVID-19 Related Queries)	KMeans (k = 99), reviewer 1				KMeans (k = 99), reviewer 2			
	Round1	Round2	Round3	Mean	Round1	Round2	Round3	Mean
Was patient in contact with a person with known COVID-19?	0.60	1.00 (20/20)	1.00 (20/20)	0.87	0.65	0.95 (19/20)	1.00 (20/20)	0.87
Does patient have a known exposure?	0.95	1.00 (20/20)	1.00 (20/20)	0.98	0.95	1.00 (20/20)	1.00 (20/20)	0.98
Does patient have a fever?	1.00	1.00 (20/20)	1.00 (20/20)	1.00	1.00	1.00 (20/20)	1.00 (20/20)	1.00
Does patient have shortness of breath?	1.00	1.00 (20/20)	1.00 (20/20)	1.00	0.95	1.00 (20/20)	1.00 (20/20)	0.98
Does patient have a cough?	0.95	1.00 (20/20)	1.00 (20/20)	0.98	0.95	1.00 (20/20)	1.00 (20/20)	0.98
Is patient having difficulty breathing?	0.70	1.00 (20/20)	1.00 (20/20)	0.90	0.75	1.00 (19/19)	1.00 (19/19)	0.92
Does patient have a recent COVID test?	0.10	0.76 (13/17)	1.00 (14/14)	0.62	0.40	1.00 (20/20)	1.00 (20/20)	0.80
Has patient tested positive for COVID?	1.00	1.00 (20/20)	1.00 (20/20)	1.00	1.00	1.00 (20/20)	1.00 (20/20)	1.00
When was the patient's COVID test?	0.05	0.83 (10/12)	1.00 (10/10)	0.63	0.10	0.58 (7/12)	1.00 (9/9)	0.56
Overall MAP	0.71	0.95	1.00	0.89	0.75	0.95	1.00	0.90

pulmonary hypertension with elevated filling pressure' can be retrieved as a response for the query 'Does patient have high blood pressure'.

We noticed that for queries like 'Did patient have a fall?', the performance of Word2Vec is worst because the model considers 'fall' as synonyms of 'decline' or 'decrease', rather than 'lose one's balance and collapse' in the first round. This is due to the fact that 'fall' is often used in the same context as 'decrease'. Also, for the query 'Does patient have a known exposure?', Word2Vec works much worse than clinicalBERT. In this case, Word2Vec retrieves information like 'Exposure Details: Never smoker' because Word2Vec cannot fully understand context due to limited window size during training phase (see supplementary Tables 2 and 3 for detailed results for individual general and COVID-19 specific queries respectively). For queries like -'Is patient diabetic?', Word2Vec works much better than clinicalBERT because clinicalBERT also retrieves diaphoretic and diuretics related information. The performance of TF-IDF is unstable. While it performs much worse than clinicalBERT on general queries, it performs better than clinicalBERT for COVID-19 specific queries in the first round and only slightly worse in the final round (0.99 MAP for TF-IDF vs. 1.00 MAP for clinicalBERT). The reason behind such unstable performance is that the model discards some low-frequency tokens in order to optimize the length of TF-IDF sentence embedding for reducing sparseness, leaving it unable to properly

Table 3
Precision for general queries across rounds 1, 2, and 3 for pseudo relevance feedback.

Pseudo Relevance Feedback (General Queries)	Round1	Round2	Round3	Mean
Does patient have any surgeries?	0.8	1.00 (16/16)	1.00 (16/16)	0.93
What is surgical history of the patient?	0.10	0.00 (0/1)	0.00 (0/1)	0.03
Does patient have diabetes?	0.25	1.00 (5/5)	1.00 (5/5)	0.75
Is patient diabetic?	0.55	1.00 (11/11)	1.00 (11/11)	0.85
Does patient have hypertension?	0.35	0.73 (8/11)	1.00 (10/10)	0.69
Does patient have high blood pressure?	0.05	0.22 (2/9)	0.17 (1/6)	0.15
Does patient have diabetes and hypertension?	0.55	1.00 (11/11)	1.00 (11/11)	0.85
Has patient had a stroke?	0.15	0.27 (3/11)	0.75 (3/4)	0.39
Does patient history of stroke?	0.05	0.25 (2/8)	1.00 (1/1)	0.43
Did patient have a fall?	0.85	1.00 (18/18)	1.00 (18/18)	0.95
Is patient hypoxic?	1.00	1.00 (20/20)	1.00 (20/20)	1.00
Is the patient's SpO2 ≤ 94% on room air?	1.00	1.00 (20/20)	1.00 (20/20)	1.00
Is the patient requiring supplemental oxygen?	0.55	0.80 (12/15)	0.86 (12/14)	0.74
Does the patient have an abnormal chest X-ray?	0.90	1.00 (18/18)	1.00 (18/18)	0.97
Is the patient requiring mechanical ventilation?	0.60	0.57 (8/14)	1.00 (11/11)	0.72
When did the patient's symptoms start?	0.10	0.37 (7/19)	1.00 (7/7)	0.49
When was the patient's last hospitalization?	0.10	0.23 (3/13)	1.00 (4/4)	0.44
Overall MAP	0.47	0.67	0.87	0.67

Table 4
Precision for COVID-19 specific queries across rounds 1, 2, and 3 for pseudo relevance feedback.

Pseudo Relevance Feedback (COVID-19 Related Queries)	Round1	Round2	Round3	Mean
Was patient in contact with a person with known COVID-19?	0.60	1.00 (20/20)	1.00 (20/20)	0.87
Does patient have a known exposure?	0.95	1.00 (20/20)	1.00 (20/20)	0.98
Does patient have a fever?	1.00	1.00 (20/20)	1.00 (20/20)	1.00
Does patient have shortness of breath?	1.00	1.00 (20/20)	1.00 (20/20)	1.00
Does patient have a cough?	0.95	1.00 (20/20)	1.00 (20/20)	0.98
Is patient having difficulty breathing?	0.70	1.00 (19/19)	1.00 (15/15)	0.90
Does patient have a recent COVID test?	0.10	0.33 (2/6)	0.88 (7/8)	0.44
Has patient tested positive for COVID?	1.00	1.00 (20/20)	1.00 (20/20)	1.00
When was the patient's COVID test?	0.05	0.00 (0/7)	0.00 (0/1)	0.02
Overall MAP	0.71	0.81	0.88	0.80

represent some queries. For example, 'Is the patient diabetic?' achieves 0.00 MAP because 'is' and 'patient' are stopwords, and 'diabetic' is discarded when using the TF-IDF model because of its relatively low frequency in our dataset. Besides, the effect of user feedback is not clear for TF-IDF for general queries.

Table 5
Pairwise student's *t*-test of the precision between initial retrieval and two rounds of relevance feedback. The Null hypothesis (H_0) assumes precision between initial retrieval and two rounds of relevance feedback are identical.

Reviewers	Measure	p-value	Confidence level
Reviewer 1	round1-round2	0.011091	H_0rejected with confidence level > 95%
	round1-round3	0.002530	H_0rejected with confidence level > 99%
Reviewer 2	round1-round2	0.006837	H_0rejected with confidence level > 99%
	round1-round3	0.000019	H_0rejected with confidence level > 99%

4. Discussion

Extraction of relevant information from the EHR is often difficult and time-consuming. Each patient may have thousands of records often with multiple events occurring on the same day[24]. Although these records are rich in information, they are variable in quality and may be missing or incomplete. In this work, we build a query bot to support real time retrieval of clinical history from clinical notes of COVID-19 patients using BERT sentence embedding using natural language queries. The proposed querybot platform retrieves clinical history of patients using *natural language query* and demonstrate the utilization for generic as well as COVID-19 specific queries. By incorporating the user feedback mechanism, our system allows too modify the search space based on user preferences and makes patient clinical history accessible to different types of users, including those with little expertise or those with little understanding of the underlying data models and/or nuanced coding schemes. We also apply a clustering method to reduce computation time for near real-time querying and build an interface for collecting user feedback to improve performance. The precision reached > 0.9 for most of the 17 general queries with a MAP can reach 0.93–0.94 after relevance feedback, and the precision reached 1.00 for most of the 9 COVID-19 related queries. This trend is clearly indicated by the performance reported in Tables 1 and 2 for general and COVID-19 related queries respectively. Besides, given the peculiarity of the queries, the retrieved results of COVID-19 specific queries are good even without user feedback, and are obviously better with feedback than that of general queries. The proposed model uses the Rocchio algorithm for iterative incorporation of user feedback. Comparative evaluation of the model with pseudo relevance feedback clearly indicates the advantages of our design decisions regarding incorporation of user feedback. As can be seen by comparing Tables 1 and 3, and Tables 2 and 4, the results of the Rocchio algorithm for both general queries and COVID-19 related queries are better than the results of pseudo relevance feedback. MAP values for users 1 and 2 for Rocchio algorithm, and pseudo relevance feedback over three rounds, as shown Fig. 6, clearly indicate the overall improvement achieved by using multiple feedback rounds. While pseudo relevance feedback is able to improve the performance over multiple rounds, performance gain achieved by Rocchio outperforms pseudo relevance by a clear margin in all cases. p-values reported in Table 5 provide further evidence of the performance gain in terms of precision scores achieved by incorporating iterative user relevance feedback in the proposed model. More importantly, the null hypothesis is rejected with higher statistical significance ($p \ll 0.01$) when the baseline round 1 is compared with the round 3, as compared to round 2 of feedback. Besides, we compared clinicalBERT with other popular word embedding methods, like Word2Vec and TF-IDF, as shown in Tables 6 and 7. The MAP values for Word2Vec in the first and second rounds for general queries are worse than those of clinicalBERT, though it outperforms clinicalBERT in the third round. For COVID-19 specific queries, Word2Vec achieves better MAP value than clinicalBERT in the first round but performs worse than clinicalBERT in later rounds. The

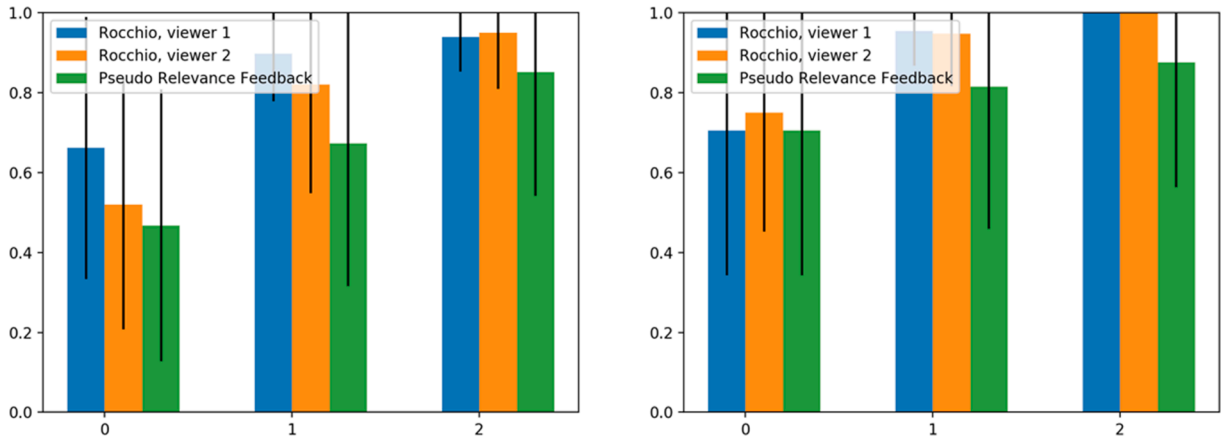

Fig. 6. Average MAP and standard deviation for model performance over rounds 1,2, and 3 for Rocchio algorithm and Pseudo Relevance Feedback, demonstrating an increase in performance with incorporation of user feedback. Results for general queries (left) and COVID-19 related queries (right) are shown.

Table 6
MAP of Word2Vec, TF-IDF, bioBERT, and clinicalBERT for general queries across rounds 1, 2, and 3 for reviewer 1 of Rocchio algorithm.

Rocchio Algorithm (General Queries)	Overall MAP			
	Round1	Round2	Round3	Mean
Word2Vec, reviewer 1	0.55	0.85	0.96	0.79
TF-IDF, reviewer 1	0.44	0.48	0.51	0.48
bioBERT, reviewer 1	0.57	0.72	0.84	0.71
clinicalBERT, reviewer 1	0.66	0.90	0.94	0.83

Table 7
MAP of Word2Vec, TF-IDF, bioBERT, and clinicalBERT for COVID-19 related queries across rounds 1, 2, and 3 for reviewer 1 of Rocchio algorithm.

Rocchio Algorithm (COVID-19 Related Queries)	Overall MAP			
	Round1	Round2	Round3	Mean
Word2Vec, reviewer 1	0.73	0.86	0.93	0.84
TF-IDF, reviewer 1	0.74	0.93	0.99	0.89
bioBERT, reviewer 1	0.74	0.83	0.90	0.82
clinicalBERT, reviewer 1	0.71	0.95	1.00	0.89

clinicalBERT performed better than both Word2Vec and TF-IDF considering the balance between precision and the capacity to learn from user-feedback. We also compared clinicalBERT with bioBERT [25], (which is a specific medical domain version BERT pre-trained on PubMed abstracts (PubMed) and PubMed Central full-text articles (PMC) for 200 k steps), as shown in Table 7 and Table 8. The MAP for bioBERT is 0.57 in the first round and 0.84 in the final round. The performance of clinicalBERT is better than bioBERT because clinicalBERT is fine-tuned on MIMIC notes, which are also clinical notes as our data.

In addition to quantitative measures, we visualize the sentence and query embeddings of the first two separate retrieval rounds in order to qualitatively evaluate the importance of the feedback. Fig. 4 demonstrates the relevance feedback for two queries: '*What is surgical history of the patient?*' and '*Has patient had a stroke?*', projected in 2D space using t-SNE. After feedback, the query vectors move towards the cluster of relevant sentences in the language space, indicating more meaningful results after relevance feedback. The proposed system can offer the flexibility of formulating natural language queries and retrieving clinical history of the patient's real time which may improve timeliness of critical information and ultimately help the patient management. Our system is light-weighted and can be easily operated and handled at any clinical facility. Demo video of the system can be found here – https://youtu.be/GYpMBGHy080, and we made the training code publicly available in https://github.com/YiboWANG214/QueryBot_COVID19. We are currently planning to test the system in a clinical setting with and without clinician feedback to support COVID-19 patient-care as a quality and research application at Emory clinic.

Our work has several limitations. First, the performance of our model is sensitive to user feedback, which means the performance may vary across users and is sensitive to incorrect feedback. Due to ambiguity of natural language record documentation, the relevance of retrieved sentences may be judged as vague by the expert (see supplementary Table 1 for sample vague queries). Saving the modified language space for each user could be a feasible solution to enhance the performance for individual users over time. The second limitation is that clustering causes results of some queries to be unsatisfactory. In fact, this is a trade-off between computation time and model precision. Despite these limitations, we believe our work shows the potential value for applying BERT and relevance feedback to clinical notes information retrieval both for clinical and research purposes. In future work, we will test our system with more users and more diverse queries and aim to improve clustering accuracy to maintain speed while increasing precision.

Authors contribution

Yibo Wang developed the retrieval system and performed the analysis. Amara Tariq contributed in problem formulation and data analysis. Fiza Khan, Judy Wawira Gichoya, and Hari Trivedi trained the readers and contributed in system validation. Imon Banerjee planned the system, analyzed the data and performed the analysis. All the authors contributed in writing and revising the manuscript.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.jbi.2021.103918.

References

[1] Vijayvargiya P, Esquer Garrigos Z, Castillo Almeida NE, Gurram PR, Stevens RW, Razonable RR. Treatment Considerations for COVID-19. Mayo Clinic Proceedings 2020;95(7):1454-66 doi: 10.1016/j.mayocp.2020.04.027published Online First: Epub Date]|.
[2] Gillis J. The History of the Patient History since 1850. Bulletin of the History of Medicine 2006;80(3):490-512 doi: 10.1353/bhm.2006.0097published Online First: Epub Date]|.
[3] Xu H, Stetson PD. A Study of Abbreviations in Clinical Notes.5.

[4] Luo Y-F, Sun W, Rumshisky A. A Hybrid Normalization Method for Medical Concepts in Clinical Narrative using Semantic Matching.9.
[5] Robertson S, Zaragoza H. The Probabilistic Relevance Framework: BM25 and Beyond. FNT in Information Retrieval 2009;3(4):333-89 doi: 10.1561/1500000019published Online First: Epub Date]|.
[6] Harris ZS. Distributional Structure. WORD 1954;10(2-3):146-62 doi: 10.1080/00437956.1954.11659520published Online First: Epub Date]|.
[7] Salton G, Wong A, Yang CS. A vector space model for automatic indexing. Commun. ACM 1975;18(11):613-20 doi: 10.1145/361219.361220published Online First: Epub Date]|.
[8] Liu T-Y. Learning to Rank for Information Retrieval. FNT in Information Retrieval 2007;3(3):225-331 doi: 10.1561/1500000016published Online First: Epub Date]|.
[9] W. Gregg, J. Jirjis, N.M. Lorenzi, D. Giuse, StarTracker: an integrated, web-based clinical search engine, AMIA Annu Symp Proc 855 (2003).
[10] D.A. Hanauer, EMERSE: The Electronic Medical Record Search Engine, AMIA Annu Symp Proc 941 (2006).
[11] Wei X, Eickhoff C. Embedding Electronic Health Records for Clinical Information Retrieval. arXiv:1811.05402 [cs] 2018.
[12] Yong R, Huang TS, Ortega M, Mehrotra S. Relevance feedback: a power tool for interactive content-based image retrieval. IEEE Trans. Circuits Syst. Video Technol. 1998;8(5):644-55 doi: 10.1109/76.718510published Online First: Epub Date]|.
[13] Ian R, Mounia L. A survey on the use of relevance feedback for information access systems. Knowl. Eng. Rev. 2003;18(2):95–145 doi: 10.1017/s0269888903000638published Online First: Epub Date]|.
[14] Drucker H, Shahrary B, Gibbon DC. Relevance Feedback using Support Vector Machines.8.
[15] Bachman J. Reverse-Transcription PCR (RT-PCR). Methods in Enzymology: Elsevier, 2013:67-74.
[16] 1. Devlin J, Chang M-W, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding; 2019; Minneapolis, Minnesota. Association for Computational Linguistics.
[17] Alsentzer E, Murphy J, Boag W, Weng WH, Jindi D, Naumann T, McDermott M. Publicly Available Clinical BERT Embeddings. In Proceedings of the 2nd Clinical Natural Language Processing Workshop 2019 Jun (pp. 72-78).
[18] Zhang L, Zhao L, Qin S, Pfoser D. TG-GAN: Deep Generative Models for Continuously-time Temporal Graph Generation. ArXiv 2020;abs/2005.08323.
[19] Reimers N, Gurevych I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. arXiv:1908.10084 [cs] 2019.
[20] M. Laurens van der, H. Geoffrey, Visualizing Data using t-SNE, J. Machine Learning Res. 9 (2008) 2579–2605.
[21] Board PDQPTE. Osteosarcoma and Undifferentiated Pleomorphic Sarcoma of Bone Treatment (PDQ®): Health Professional Version. PDQ Cancer Information Summaries. Bethesda (MD): National Cancer Institute (US), 2002.
[22] H. Phil, Python and Tkinter Programming, Linux J. (2000), 2000(77es):23–es.
[23] M.D. Smucker, J. Allan, B. Carterette, A comparison of statistical significance tests for information retrieval evaluation, ACM Press, 2007 2007..
[24] Wang Y, Wang L, Rastegar-Mojarad M, et al. Clinical information extraction applications: A literature review. J Biomed Inform 2018;77:34-49 doi: 10.1016/j.jbi.2017.11.011published Online First: Epub Date]|.
[25] Lee J, Yoon W, Kim S, et al. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics 2019;36(4):1234-1240 doi: 10.1093/bioinformatics/btz682published Online First: Epub Date]|.